

## **BAB II**

### **LANDASAN TEORI**

Teori yang mendasari atau menjadikan landasan teori pelaksanaan dan pengerjaan skripsi ini diambil dari beberapa sumber, termasuk juga dalam hal ini penerapan kemampuan dan pengetahuan mahasiswa baik yang telah diperoleh selama duduk dibangku perguruan tinggi atau pengetahuan umum lainnya. Dengan ini mahasiswa diharapkan mempunyai mutu dan pelajaran ilmiah yang bermanfaat bagi semua pihak. Adapun landasan teori yang mendasari pelaksanaan skripsi adalah sebagai berikut.

#### **2.1. Futsal**

Futsal (futbol dalam bahasa Spanyol yang berarti sepak bola dalam ruangan) merupakan permainan sepak bola yang dilakukan dalam ruangan. Olahraga ini sendiri dilakukan oleh lima pemain setiap tim berbeda dengan sepak bola konvensional dengan ukuran bola yang lebih kecil dan aturan permainan yang begitu ketat dari sepak bola. Selain itu, futsal adalah olahraga beregu yang sangat cepat dan dinamis (Laksana, 2011). Sedangkan menurut Murhanto futsal adalah permainan bola yang dimainkan oleh dua regu, yang masing-masing beranggotakan lima orang. Tujuannya adalah memasukkan bola ke gawang lawan, dengan memanipulasi bola dengan kaki (Murhananto, 2006) (Pamungkas, 2013).

#### **2.2. Teknik Dasar Futsal**

Teknik dasar dalam permainan futsal sama dengan teknik dasar permainan sepakbola. Teknik-teknik yang digunakan dalam permainan futsal relatif tidak jauh berbeda dalam permainan sepakbola namun karena faktor

lapangan yang relatif kecil dan permukaan lantai yang lebih rata menyebabkan perbedaan-perbedaan penggunaan teknik. Teknik adalah cara melakukan atau melaksanakan sesuatu untuk mencapai tujuan tertentu secara efektif dan efisien. Menurut (Laksana, 2011) modern futsal adalah permainan futsal yang para pemainnya diajarkan bermain dengan sirkulasi bola yang sangat cepat, menyerang dan bertahan, dan juga sirkulasi pemain tanpa bola ataupun *timing* yang tepat. Oleh karena itu, diperlukan kemampuan menguasai teknik dasar bermain futsal yang meliputi:

1. Teknik Mengumpan Bola (*Passing*)

*Passing* merupakan salah satu teknik dasar permainan futsal yang sangat dibutuhkan oleh pemain. Di lapangan yang rata dan ukuran lapangan yang kecil dibutuhkan *passing* yang keras dan akurat karena bola yang meluncur sejajar dengan tumit pemain. Ini di sebabkan hampir sepanjang permainan futsal menggunakan *passing*. Untuk menguasai *passing* diperlukan penguasaan gerakan sehingga sasaran yang diinginkan tercapai (Laksana, 2011).

2. Teknik Kontrol Bola (*Control*)

Teknik dasar dalam keterampilan *control* (menahan bola) haruslah menggunakan telapak kaki (*sole*). Dengan permukaan lapangan yang rata, bola akan bergulir cepat sehingga para pemain harus dapat mengontrol dengan baik. Apabila menahan bola jauh dari kaki, lawan akan mudah merebut bola (Laksana, 2011).

### 3. Teknik Menggiring Bola (*Dribbling*)

Teknik *dribbling* merupakan keterampilan penting dan mutlak harus dikuasai oleh setiap pemain futsal. *Dribbling* merupakan kemampuan yang dimiliki setiap pemain dalam menguasai bola sebelum diberikan kepada temannya untuk menciptakan peluang dalam mencetak gol (Laksana, 2011).

### 4. Teknik Menembak (*Shooting*)

*Shooting* merupakan teknik dasar yang harus dikuasai oleh setiap pemain. Teknik ini merupakan cara untuk menciptakan gol. Ini disebabkan seluruh pemain memiliki kesempatan untuk menciptakan gol dan memenangkan pertandingan atau permainan. *Shooting* dapat dibagi menjadi dua teknik, yaitu *shooting* menggunakan punggung kaki dan ujung sepatu atau ujung kaki (Laksana, 2011).

## 2.3. Metode SMART ( Simple Multi – Attribut Rating Technique)

*SMART* (*Simple Multi – Attribute Rating Technique* ) merupakan metode pengambilan keputusan multi kriteria yang dikembangkan oleh Edward pada tahun 1977. Teknik pengambilan keputusan multi kriteria ini didasarkan pada teori bahwa setiap alternatif terdiri dari sejumlah kriteria yang memiliki nilai – nilai dan setiap kriteria memiliki bobot yang menggambarkan seberapa penting ia dibandingkan dengan kriteria lain. Pembobotan ini digunakan untuk menilai setiap alternatif agar diperoleh alternatif terbaik.

*SMART* menggunakan *linear additive model* untuk meramal nilai setiap alternatif. *SMART* merupakan metode pengambilan keputusan yang fleksibel. *SMART* lebih banyak digunakan karena kesederhanaanya dalam merespon

kebutuhan pembuat keputusan dan caranya menganalisa respon. Analisa yang terlibat adalah transparan sehingga metode ini memberikan pemahaman masalah yang tinggi dan dapat diterima oleh pembuat keputusan (Yunitarini, 2013).

Model fungsi utiliti linear yang digunakan oleh *SMART* adalah seperti berikut :

$$\text{Maximize } \sum_{j=1}^k w_j \cdot u_{ij}, \forall i = 1, \dots, n$$

Di mana :

- $w_j$  adalah nilai pembobotan kriteria ke- $j$  dari  $k$  kriteria, -  $u_{ij}$  adalah nilai utility alternatif  $i$  pada kriteria  $j$ .
- Pemilihan keputusan adalah mengidentifikasi mana dari  $n$  alternatif yang mempunyai nilai fungsi terbesar.
- Nilai fungsi ini juga dapat digunakan untuk meranking  $n$  alternatif

Teknik Metode SMART Adapun teknik atau langkah-langkah dalam proses SMART, antara lain :

1. Identifikasi user yang nantinya bertanggung jawab dalam mengambil keputusan.
2. Identifikasi Permasalahan yang ada Dengan melihat akar permasalahan dan batasan-batasan yang ada agar nantinya tidak menyimpang dari tujuan yang ingin dicapai.

3. Identifikasi alternatif yang ada untuk mencapai tujuan dari sistem yang akan dibuat.
4. Identifikasi kriteria-kriteria yang akan mendukung pengambilan keputusan
5. Memberikan peringkat atau bobot untuk setiap kriteria. Pemberian peringkat atau bobot ini ditentukan oleh user dan sistem akan memberikan bobot skala default dari peringkat yang dimasukkan.
6. Memberikan penilaian setiap kriteria untuk setiap alternatif. Juga dilakukan oleh user dimana penilaian setiap kriteria dapat bersifat kualitatif maupun kuantitatif untuk setiap alternatifnya.
7. Mengembangkan *Single-Attribute Utilities* yang mencerminkan seberapa baik setiap alternatif dilihat dari setiap kriteria. Tahap ini adalah memberikan suatu nilai pada semua kriteria untuk setiap alternatif dengan nilai yang berskala 0 sampai 1.
8. Menghitung penilaian terhadap setiap alternatif

#### **Kelebihan Metode SMART**

SMART memiliki beberapa kelebihan dibandingkan dengan metode pengambilan keputusan yang lain yaitu :

1. Mungkin melakukan penambahan / pengurangan alternatif Pada metode SMART penambahan atau pengurangan alternatif tidak akan mempengaruhi perhitungan pembobotan karena setiap penilaian alternatif tidak saling bergantung

2. Sederhana Perhitungan pada metode SMART lebih sederhana sehingga tidak diperlukan perhitungan matematis yang rumit dengan pemahaman matematika yang kuat
3. Transparan Proses dalam menganalisa alternatif dan kriteria dalam SMART dapat dilihat oleh user sehingga user dapat memahami bagaimana alternatif tertentu dapat dipilih. Alasan-alasan bagaimana alternatif itu dipilih dapat dilihat dari prosedur-prosedur yang dilakukan dalam SMART mulai dari penentuan kriteria, pembobotan, dan pemberian nilai pada setiap alternatif.
4. Fleksibilitas Pembobotan Pembobotan yang dipakai di dalam metode SMART ada 3 jenis yaitu pembobotan secara langsung (*direct weighting*), pembobotan swing (*swing weighting*), pembobotan centroid (*centroid weighting*) (Budiman H. D, 2013).

#### **2.4. Contoh Kasus Penerapan Metode SMART**

Penentuan nilai alternatif pada setiap kriteria dilakukan dengan memberi nilai pada setiap kriteria berdasarkan pengaruhnya terhadap rekomendasi pemilihan merk velg. Semakin besar nilai alternatif semakin besar nilai kriteria tersebut. Total hasil masing-masing alternatif keputusan akan relatif berbeda secara nyata dan sesuai kebutuhan

Berikut adalah langkah langkah contoh kasus Penerapan Metode *Simple Multy Attribut Rating Technique* (SMART) terhadap rekomendasi pemilihan merk velg (Saputro E. , 2016).

Langkah 1 : Menentukan Jumlah Kriteria

Tabel 2.1 Contoh menentukan Kriteria

NO	Kriteria	Nama Kriteria
1	A1	Harga
2	A2	Bahan
3	A3	Warna
4	A4	Ukuran

Langkah 2 : Menentukan nilai bobot pada setiap kriteria.

Sistem secara default memberikan skala 0-100 berdasarkan prioritas yang telah diinputkan kemudian dilakukan normalisasi.

$$\text{Normalisasi} = \frac{w_j}{\sum w_j}$$

Keterangan :

$w_j$  : Bobot suatu kriteria

$\sum w_j$  : Total bobot semua kriteria.

Tabel 2.2 Contoh menentukan Bobot Kriteria

NO	Kriteria	Bobot (%)
1	A1	35
2	A2	30
3	A3	25
4	A4	10
Total		100%

Keterangan : Memberi nilai bobot pada setiap kriteria sesuai dengan prioritas kebutuhan dari contoh diatas bobot harga adalah yang paling tinggi yaitu 35%, bahan 30%, warna 25%, dan ukuran 10 % , sehingga jika semua kriteria dijumlah akan diperoleh total bobot 100%. (Saputro E. , 2016)

## Langkah 3 : Menghitung Normalisasi Bobot Kriteria

Tabel 2.3 Contoh Menghitung Normalisasi Bobot Kriteria

No	Kriteria	Normalisasi Bobot
1	Harga	0.35
2	Bahan	0.30
3	Warna	0.25
4	Ukuran	0.10

## Langkah 4 : Menghitung Nilai Utility

Hitung nilai utility untuk setiap kriteria masing-masing.

$$u_i(a_i) = 100 \frac{(C_{max} - C_{out\ i})}{(C_{max} - C_{min})} \%$$

Keterangan :

$u_i(a_i)$  : nilai utility kriteria ke-1 untuk kriteria ke-i

$C_{max}$  : nilai kriteria maksimal

$C_{min}$  : nilai kriteria minimal

$C_{out\ i}$  : nilai kriteria ke-i

Tabel 2.4 Contoh Menghitung Nilali Utility

No	Nama	Nilai
1	C Max	10
2	C Min	1

Selanjutnya mendiskripsikan nilai ke dalam kriteria yang sudah ditentukan sesuai kebutuhan.



### Nilai Utility 1

Tabel 2.5 Contoh Menghitung Nilali Utility 1

No	Kriteria	Rumus Utility
1	Harga	$a = (10 - 9) / (10 - 1) * 100$
2	Bahan	$b = (10 - 7) / (10 - 1) * 100$
3	Warna	$c = (10 - 5) / (10 - 1) * 100$
4	Ukuran	$d = (10 - 1) / (10 - 1) * 100$

### Nilai Utility 2

Tabel 2.6 Contoh Menghitung Nilali Utility 2

No	Kriteria	Rumus Utility
1	Harga	$r = (10 - 8) / (10 - 1) * 100$
2	Bahan	$s = (10 - 6) / (10 - 1) * 100$
3	Warna	$t = (10 - 4) / (10 - 1) * 100$
4	Ukuran	$u = (10 - 3) / (10 - 1) * 100$

Terdapat dua nilai utility dikarenakan akan ada 2 rekomendasi yang di outputkan oleh sistem, yaitu Rekomendasi 1 dan Rekomendasi 2.

### Langkah 5 : Menghitung Nilai Akhir

$$u(a_i) = \sum_{j=1}^m w_j u_i(a_i),$$

### Hasil Uji Coba Rekomendasi 1

Tabel 2.7 Contoh Menghitung Nilai Akhir Rekomendasi 1

NO	Kriteria	Kategori	Hasil	Total	Rekomendasi
1	A1	A	35	100	TDR
2	A2	B	30		
3	A3	c	25		
4	A4	d	10		

## Hasil Uji Coba Rekomendasi 2

Tabel 2.8 Contoh Menghitung Nilai Akhir Rekomendasi 2

NO	Krite ria	Kategori	Hasil	Total	Rekomendasi
1	A1	r	27, 22	77, 77	Willwood
2	A2	s	23 , 33		
3	A3	t	19 , 44		
4	A4	U	7 , 7 7		

Dari hasil uji coba, didapatkan kesimpulan bahwa:

1. Karena rekomendasi yang diperoleh berbeda, yaitu antara rekomendasi 1 dan rekomendasi 2, maka dapat disimpulkan bahwa bobot sangat mempengaruhi pada sistem ini.
2. Nilai terbesar akan menjadi prioritas, dalam kasus ini ada dua merk yaitu TDR sebagai rekomendasi pertama dan WILLWOOD sebagai rekomendasi kedua.
3. Masing-masing merk memiliki batasan nilai yaitu
  - a) Rekomendasi 1:
    - i. Range nilai velg Venom antara 0 – 26
    - ii. Range nilai velg Rossi antara 26 – 40
    - iii. Range nilai velg DBS antara 41 – 60
    - iv. Range nilai velg DID antara 61 – 70
    - v. Range nilai velg Willwood antara 71 – 80
    - vi. Range nilai velg TDR antara 81 – 100

Pembagian range pada rekomendasi 1 di atas berdasarkan hasil perkalian normalisasi dan nilai utility 1. Dan hasil akhir pada rekomendasi 1 terbagi menjadi 6, berdasarkan kebutuhan.

b) Rekomendasi 2:

- i. Range nilai velg Venom antara 0 – 29
- ii. Range nilai velg Rossi antara 30 – 45
- iii. Range nilai velg DBS antara 46 – 63
- iv. Range nilai velg DID antara 64 – 74
- v. Range nilai velg Willwood antara 75 – 85
- vi. Range nilai velg TDR antara 86 – 100

Pembagian range pada rekomendasi 2 di atas berdasarkan hasil perkalian normalisasi dan nilai utility 2. Dan hasil akhir pada rekomendasi 2 terbagi menjadi 6, berdasarkan kebutuhan. Range pada rekomendasi 1 dan rekomendasi 2 berbeda dikarenakan nilai utility 1 dan nilai utility 2 berbeda (Saputro E. , 2016).

## **2.5.Delphi 7.0**

Delphi merupakan program aplikasi database yang berbasis Object Pascal dari Borland serta memiliki kemampuan membangun aplikasi yang multi- threaded. Artinya, jika aplikasi database semakin kompleks, maka aplikasi tersebut perlu dijalankan dalam beberapa bagian dan masing- masing bagian akan menjalankan fungsi tertentu. Bagian-bagian tersebut dinamakan

threads. Dengan demikian dapat dikatakan bahwa Delphi adalah sebuah program aplikasi yang mendukung pembuatan, pemakaian, pengontrolan thread (Andi, 2004).

Borland Delphi 7.0 Delphi adalah software buatan Borland yang sangat populer, yaitu sebuah bahasa pemrograman , *language development*, aplikasi untuk membuat aplikasi. Delphi digunakan untuk membangun aplikasi windows, aplikasi grafis, aplikasi visual, bahkan aplikasi jaringan (client/server) dan berbasis internet.

Konsep dasar Pemrograman Delphi :

a. Form

Semua program *windows* memiliki form dan form adalah komponen utama dalam pengembangan aplikasi karena form merupakan induk dari semua komponen. form merupakan tempat komponen-komponen lain diletakkan, seperti *button*, *combobox*, *listbox*, atau *editbox*.

b. Properti

Sebuah objek diakses melalui propertinya. Properti dari objek Delphi mempunyai nilai yang dapat diubah pada saat aplikasi dijalankan, anda harus menuliskan sedikit program.

c. Komponen Delphi

Dukungan Delphi terhadap aplikasi database merupakan salah satu fitur kunci lingkungan pemrograman. Delphi memberikan seperangkat komponen serta pelengkap lainnya sehingga pembuatan aplikasi database menjadi lebih mudah, bahkan untuk pemula sekalipun (Andi, 2004).

## 2.6.MYSQL

MySQL atau sering dibaca dengan ejaan “mai es que el” ini merupakan sebuah database yang berbasiskan server database. Kemampuannya dalam menangani RDBMS (Relasional Database Management System) mengakibatkan database ini menjadi database yang sangat populer saat ini. Database MYSQL bersifat open source dan mampu menangani data yang sangat besar hingga ukuran Giga Byte, dengan kemampuan daya tampung data ini maka MySQL sangat cocok digunakan untuk mengcover data pada perusahaan baik yang kecil sampai perusahaan besar (Andi, 2004).

MYSQL awalnya dibuat oleh perusahaan konsultan bernama TeX yang berlokasi di Swedia. Saat ini pengembangan MySQL berada dibawah naungan perusahaan MySQL AB. software dapat diunduh di situs [www.mysql.com](http://www.mysql.com).

Sebagai software DBMS, MySQL memiliki sejumlah fitur seperti yang dijelaskan dibawah ini (Kadir, 2008).

a. Multiplatform

MySQL tersedia pada beberapa platform seperti (Windows, Linux, Unix, dan lain-lain).

b. Andal, cepat dan mudah digunakan

MySQL tergolong sebagai database server (server yang melayani permintaan terhadap database) yang andal, dapat menangani database yang besar dengan kecepatan tinggi, mendukung banyak sekali fungsi untuk mengakses database, dan sekaligus mudah untuk digunakan.

c. Jaminan keamanan akses

MySQL mendukung pengamanan database dengan berbagai kriteria

pengaksesan. Sebagai gambaran, dimungkinkan untuk mengatur user agar bisa mengakses data yang bersifat rahasia (misalnya gaji pegawai), sedangkan user lain tidak boleh.

d. Dukungan SQL

Seperti tersirat dalam namanya, MySQL mendukung perintah SQL (Structured Query Language). Sebagaimana diketahui, SQL merupakan standar dalam akses database relasional. Pengetahuan akan SQL akan memudahkan siapa pun untuk menggunakan MySQL.

## **2.7. *Software Development Life Cycle (SDLC)***

SDLC merupakan kepanjangan dari *Software Development Life Cycle*. *Software Development Life Cycle* atau sering disebut juga *System Development Life Cycle* adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat sebelumnya (A.S. dan Shalahuddin 2011). Terdapat banyak model SDLC yang dapat digunakan dalam pengembangan sistem, salah satunya adalah model *waterfall*.

Model SDCL air terjun (waterfall) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model *waterfall* menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengkodean, pengujian, dan tahap pendukung (*support*) (Mujilan, 2013).

## 2.8.UML (Unified Modeling Language)

*Unified Modeling Language* (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML mendefinisikan notasi dan syntax/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Notasi UML diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (Object- Oriented- Design), Jim Rumbaugh OMT (Object Modeling Technique) dan Ivar Jacobson OOSE (Object-Oriented Software Engineering). Sampai era tahun 1990 terdapat puluhan metodologi pemodelan berorientasi objek telah bermunculan di dunia. Diantaranya tiga notasi tersebut di atas. Masa itu dikenal dengan masa perang metodologi ( method war ) dalam pendesainan berorientasi objek. Karena setiap metodologi membawa notasi sendiri-sendiri, mengakibatkan masalah ketika bekerjasama dengan group/perusahaan lain. Mulai Oktober 1994 Booch, Rumbaugh dan Jacobson memelopori usaha untuk menyatukan metodologi pendesainan berorientasi objek. Tahun 1995 direlease draft pertama dari UML. Pada tahun 1999 ketiga tokoh tersebut menyusun tiga buku serial tentang UML, sejak saat itulah UML menjadi standar bahasa pemodelan untuk aplikasi berorientasi objek (Mujilan, 2013). Berikut daftar UML 2 terdiri dari 13 jenis diagram resmi seperti dalam tabel berikut ini: Jenis Diagram Resmi UML

Tabel 2.9 Tabel Jenis Diagram UML

<b>Diagram</b>	<b>Kegunaan</b>
<i>Activity</i>	<i>Behaviour</i> prosedural dan parallel
<i>Class</i>	<i>Class, fitur</i> , dan hubungan-hubungan
<i>Communication</i>	Interaksi antar objek; penekanan pada jalur
<i>Component</i>	Struktur dan koneksi komponen
<i>Composite structure</i>	Dekomposisi <i>runtime</i> sebuah <i>class</i>
<i>Deployment</i>	Pemindahan artifak ke <i>node</i>
<i>Interaction overview</i>	Campuran <i>sequence</i> dan <i>activity diagram</i>
<i>Object</i>	Contoh konfigurasi
<i>Package</i>	Struktur hirarki <i>compile-time</i>
<i>Sequence</i>	Interaksi antar objek; penekanan pada <i>sequence</i>
<i>State machine</i>	Bagaimana <i>even</i> mengubah objek selama aktif
<i>Timing</i>	Interaksi antar objek; penekanan pada <i>timing</i>
<i>Use case</i>	Bagaimana pengguna berinteraksi dengan sebuah sistem

UML (Unified Modeling Language) dapat digunakan untuk menjelaskan beberapa hal yang penting dalam sistem. Tidak semua diagram UML harus dipakai, dan tidak semua sistem dijelaskan dengan UML karena terdapat model penjelasan lain, atau dengan kata lain harus dipilih yang penting dalam penjelasan sistem sesuai kemampuan perancang dan pemahaman penulis program .

Bagi perancang pemula disarankan untuk memusatkan perhatian pada bentuk-bentuk diagram dasar class diagram dan sequence diagram, diagram tersebut adalah bentuk yang paling umum dan diagram yang paling berguna. Class diagram mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat diantara mereka. Class diagram juga menunjukkan properti dan operasi sebuah class dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut. UML menggunakan istilah fitur sebagai istilah umum yang meliputi properti dan operasi sebuah class .



Sequence diagram , secara khusus, menjabarkan behavior sebuah skenario tunggal. Diagram tersebut menunjukkan sejumlah objek contoh dan pesan-pesan yang melewati objek-objek di dalam use case (Mujilan, 2013).